

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Arijan Ljoki

Mobilna aplikacija za parkiranje

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.¹

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Arijan Ljoki, z vpisno številko **63090238**, sem avtor diplomskega dela z naslovom:

Mobilna aplikacija za parkiranje

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 19. septembra 2014

Podpis avtorja:

*Iskreno se zahvaljujem mentorju doc. dr. Roku Rupniku in moji družini
za vzpodbudo in čas.*

Vsem mojim.

kratica	pomen
MSDN	Microsoft Developers Network
WP	Windows Phone
XML	eXtensible Markup Language
UI	User Interface
SDK	Software Developers Kit
XAML	eXtensible Application Markup Language
URL	Uniform Resource Locator

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Primer uporabe	3
3	Uporabljena programska orodja in tehnologija	5
3.1	Adobe Photoshop	6
3.2	Windows Phone	7
3.3	Visual Studio	10
3.4	XAML	11
4	Aplikacija	13
4.1	Dostop do podatkov	13
4.2	Aplikacija	17
4.3	Mapa	19
4.4	Trola	23
4.5	Parkirišča	25
4.6	Parkirne hiše	28
4.7	Vključene knjižnice	28
4.8	Testiranje	29
4.9	Deljenje aplikacije	29

5 Sklepne ugotovitve	31
----------------------	----

Povzetek

V sklopu diplomske naloge je bila razvita mobilna aplikacija za Windows Phone platformo.

Mobilni telefoni zadnja leta omogočajo veliko več funkcij kot so klicanje in pošiljanje kratkih sporočil. Z njimi si na veliko načinov lajšamo življenje. Če smo v novem kraju in nas zanima, kje si lahko ogledamo lokalne zanimivosti, lahko si prevedemo tekst, da vemo kaj smo naročili. Lahko preverimo kdaj pride naslednji (mestni) avtobus. Veliko problemov nam lahko pomaga telefon rešit, če smo v neznanem kraju. Kot so navigacija do iskanega naslova, omejitve in prepovedi.

V aplikaciji je združenih več funkcionalnosti, ki uporabniku olajšajo napore z iskanjem parkirišč, informacijo o njih in tudi podatke o mestnem prevozu. Cilj je, da lahko uporabnik, glede na svojo lokacijo poišče najbližjo parkirno hišo, kjer lahko preveri cene in razpoložljivost ter delovni čas. Lahko pa tudi poišče parkirno hišo, glede na lokacijo, ki jo mora obiskat v novem in nepoznanem kraju. Aplikacija ga odpelje do izbranega parkirišča in si shrani njegovo lokacijo, tako da ga lahko naknadno vodi tudi nazaj. Prav tako izračuna, koliko je potrebno plačati, glede na čas parkiranja in urno postavko.

Ključne besede: mobilna aplikacija, Windows Phone, parkirišča

Abstract

For my thesis I created a mobile application for Windows Phone platform. Title of my thesis is Mobile application for parking and it is a system, that connects car parks in Slovenia.

Mobile phones are no longer used only for calling and sending short text messages. They help us on daily basis. If we are in new and unknown area and we want to know where we can see local interesting spots, they can translate text for us, so we know what to order in restaurant. We can check when next bus is coming, because we just saw one leave in front of us. There are a lot of problems which our smart phones help us solve. Like navigation to wanted address, restrictions and rules.

In application there are more than one functionalities, which help user with problem of finding right and free parking space, with information about them and data about city bus. The point of this application is that user can find car park based on location, where he can check prices and available free parking spaces. Application then navigates him to the car park he chose and saves his location so he knows where to return for his car. When he wants to leave, the application will calculate how much he needs to pay by how many hours he was there and how much is an hour.

Keywords: Mobile app, Windows Phone, car park

Poglavje 1

Uvod

V današnjem času si je skoraj nemogoče predstavljati vsakdanjika brez uporabe pametnega telefona. Že zdavnaj je uporaba pametnih mobilnikov prešla mejo, ko je uporabnik lahko telefon uporabljal samo za opravljanje klicev in pošiljanje kratkih sporočil. Današnji mobilni aparati so primerni za opravljanje bolj zahtevnih operacij, denimo pregled elektronske pošte, brskanje po internetu, igranje igrice in tudi uporaba že bolj zahtevnih programov. Po pregledu trga sem opazil, da programa, ki bi pomagala uporabniku pri iskanju praznega parkirnega prostora ni. Iz tega razloga sem se tudi odločil izdelati aplikacijo, ki si jo lahko uporabnik brezplačno naloži na pametni mobilni telefon.

Aplikacija pomaga uporabniku, da samodejno poišče prazno parkirno mesto in preveri število praznih parkirnih mest in ceno parkirne ure. Program vsebuje predhodno shranjene lokacije parkirnih hiš. Ostale podatke pridobi iz spletne strani, med drugim tudi informacije o mestnih avtobusih.

Program je razvit za Windows Phone platformo in ni kompatibilen s konkurenčnimi ponudniki operacijskih sistemov. Trenutno je potrebno lokacije parkirnih hiš vnesti ročno. Ob razširitvi uporabe programa na celotno področje Slovenije, bi lahko ustvaril še spletno stran, kjer bi se lahko vpisali lastniki parkirišč ter vnesli podatke o svojih parkirnih hišah, ki bi se nato prikazale v aplikaciji.

Poglavje 2

Primer uporabe

Voznik ob prihodu v neznano mesto ne ve, kje lahko varno pusti svoje vozilo. S to aplikacijo si lahko že naprej pregleda lokacije ter zasedenost parkirnih hiš, ki sodelujejo v tem projektu in so vneseni v aplikaciji. S klikom na parkirno hišo, ki jo pokaže na zemljevidu, vidi njene podrobnosti, kot so zasedenost parkirnih mest, čas obratovanja in cenik za eno uro ter koliko minut je zastoj. Ko se odloči za parkirno hišo, klikne na njo in se mu odpre okno, kjer ima na voljo navigacijo do parkirne hiše od svoje trenutne lokacije, ki ga nato vodi do izbrane. Nato izbere začetek in se lokacija parkirišča shrani na mapo, ki mu označi, kje je parkirano njegovo vozilo, da se zna vrniti po njega. Hkrati pa mu sešteva čas, in sicer koliko časa je bil parkiran in na koncu izračuna, glede na urno postavko, koliko mora plačati na avtomatu. Ko se vrne po vozilo, klikne izračun in sledi navodilom na zaslonu. Za zaključek plačila mora prisloniti telefon k avtomatu in pritisniti »pokliči«, kjer se sestavi številka, ki jo je potrebno poklicati.

Aplikacija ima tudi možnost iskanja, kdaj pride mestni avtobus na izbrano postajo. Z vnosom imena postaje aplikacija na spletni strani preveri, kdaj pridejo naslednji tri avtobusi in jih izpiše na zaslon. Podatki so pridobljeni iz spletne strani www.trola.si in so do minute natančni.

Poglavje 3

Uporabljen programski orodja in tehnologija

Aplikacija je razvita v Microsoft Visual Studio Express za Windows Phone. Brezplačna je za vse, ki želijo razvijati aplikacije in je na voljo na njihovi spletni strani. Uporabljali smo še Photoshop za urejanje slik.

Aplikacija je sestavljena iz dveh delov, XAML in C#. V XAML so definirane knjižnice in oblikovanje aplikacije. Postavitev gradnikov in izgled. V ozadju je koda napisana v C# jeziku.

Aplikacija je razvita za Windows Phone različice od 7.5 naprej.

Testiranje aplikacije smo delali na emulatorju, ki smo ga pridobil iz Microsoftove spletne strani. Znotraj je na voljo več različnih mobilnikov, s katerimi smo testirali, kako deluje aplikacija. Mentor na fakulteti je za testiranje aplikacije na terenu dal v uporabo mobilni telefon z Windows Phone platformo HTC 8X. S tem smo ugotavljali delovanje lokacij in samo funkcionalnost aplikacije.



Slika 3.1: Adobe Photoshop CS6 verzija.

3.1 Adobe Photoshop

Adobe PhotoShop je najboljše orodje za obdelavo fotografij. Razvilo ga je podjetje Adobe Systems, ki je vodilni izdelovalec grafičnih programov na svetu. Omogoča urejanje slik in manipuliranje z njimi. Za to orodje sem se odločil, ker sem ga že prej uporabljal in mi je poznan. Tako smo lahko na hitro uredili slike, da so bile takšne, kot smo si jih zamislili. Uporabljali smo trial verzijo, saj je drugače potrebno plačati za licenco.

Uporabljali smo Photoshop CS6, ki je izšel maja 2012. Ima veliko izboljšav glede zmogljivosti in dodatkov, ki jih prej ni bilo. Dodatno je shranjevanje v ozadju, ki omogoča uporabniku, da ureja drugo sliko, dokler se prvi projekt shranjuje. Omogoča tudi samodejno shranjevanje, kar pomeni, da če po pomoti zapremo program in slike ne shranimo, to naredi program sam.



Slika 3.2: Windows Phone 8.1

3.2 Windows Phone

Windows Phone je platforma za mobilne telefone, ki ga je razvilo podjetje Microsoft. Na začetku jim sicer ni šlo najbolje, saj sta bila glavna konkurenta Android in iOS preveč razširjena in premočna konkurenta. Microsoft je nato kupil Nokia, ki je sedaj vodilni mobilni telefon s to platformo. Vsak mesec je Windows Phone bolj in bolj razširjen, saj narašča število uporabnikov, poleg tega pa tudi redno prihajajo ven nove posodobitve in izboljšave, s čimer pa dohiteva konkurenco.

Njegov izgled je drugačen od konkurenčnih platform. Ponuja minimalističen izgled, kjer so ikone ploščice na zaslonu, ki se jim reče Live Tiles. Te ploščice omogočajo prikaz imena aplikacije ter se lahko spreminjajo v ozadju. Brez odpiranja aplikacije se ploščice osvežijo in prikažejo potrebovan podatek. Vsaka ploščica je lahko v treh različnih velikostih, od tega pa je tudi odvisno, koliko informacij lahko prikaže. Ozadje ploščice je lahko barva ali slika, tako je telefon za uporabnike bolj osebni in prilagodljiv.

Nalaganje aplikacij je možno samo preko njihove trgovine Store. Da lahko nalagamo aplikacije na telefon, je potrebno le-tega najprej odkleniti. S tem je poskrbljeno za varnost, saj preprečuje okuženje telefona s kakšno nezaželeno aplikacijo.

Nove ustvarjene aplikacije je potrebno naložiti preko računa, ki ga registriramo na Microsoftovi strani. Račun je, razen za študente, plačljiv. Pred odobritvijo aplikacije v trgovini, jo testirajo na več različnih telefonih ter preverijo, če izkorišča podatke na nedovoljen način. Aplikacijo dodajo v spletno trgovino, če je brez napak, v nasprotnem primeru pa napišejo razlog zavrnitve.

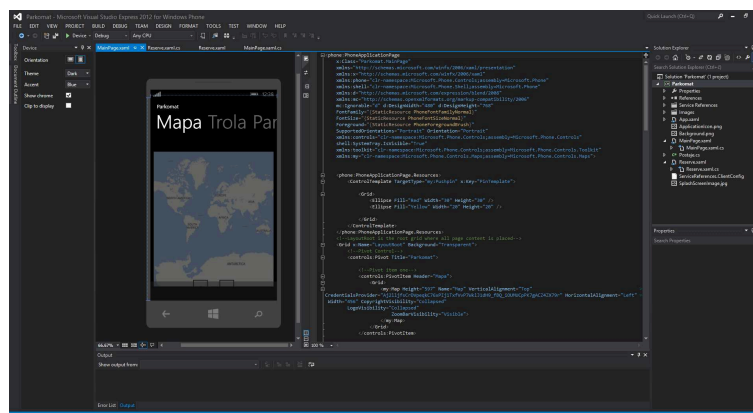
Ena večjih posodobitev, ki je razveselila uporabnike Windows Phone mobilnih telefonov, je Cortana. Gre za osebno asistentko, kateri lahko narekujemo naše želje, npr. nastavitev budilke in opomnika, pisanje kratkih sporočil in drugo. Ker jo še izboljšujejo, je v beta verziji, vseeno pa ponuja že veliko stvari, ki nam olajšajo uporabo telefona. Cortana je na voljo samo za večje države, saj je vezana na jezik telefona, zato je v Sloveniji ne moremo uporabljati. Pri testiranju smo nastavili telefon na angleški jezik in je delovalo. Zanimiva stvar, ki jo zmora je, da išče podatke po spletu, npr. vremenska napoved, letalske karte, novice.

Prvi mobilni telefon s platformo Windows Phone je izšel novembra 2010. Telefon deluje tekoče in brez problemov ter je enostaven za uporabo. Predvsem pa je povezan z operacijskim sistemom Windows 8.

Trenutno je na trgu že Windows Phone 8, ki je zelo povezan z operacijskim sistemom Windows 8. Z menjavo teme na telefonu, se zamenja tudi na računalniku. Opomniki, ki so shranjeni v koledarju se prikažejo tudi na računalniku ter obvestijo uporabnika o času dogodka.

Zahteve, ki so bile na začetku zelo strogo določene, se z novimi verzijami malo rahljajo, saj so opustili tudi zahteve glede obveznih gumbov na napravi. Zadnje minimalne zahteve (za Windows Phone 8.1) trenutno so: - Qualcomm Snapdragon S4, Snapdragon 200, Snapdragon 400, ali Snapdragon 800 večjedrni procesor - najmanj 512 Mb delovnega pomnilnika (1Gb za za-

slone z ločljivostjo 720p/1080p) - najmanj 4Gb flash pomnilnika - GPS in A-GPS - micro-USB 2.0 podpora - kamera zadaj (vsaj VGA ločljivosti) - merilnik pospeška - WiFi 802.11b/g in Bluetooth - vsaj 4-točkovni kapacitivni zaslon



Slika 3.3: Visual Studio Express for Windows Phone

3.3 Visual Studio

Visual Studio Express za Windows Phone je razvojno orodje podjetja Microsoft. Za razvijalce je zastonj na njihovi spletni strani, prav tako tudi dodatki za testiranje aplikacij. Emulator, ki je priložen omenjenemu programu, je namenjen testiranju aplikacije na različnih platformah za namen preizkusa odzivnosti in delovanja. To razvojno orodje je namenjeno samo za razvoj aplikacij za mobilne telefone. Program je dobro narejen in ob pisanju kode sam ponuja smiselne zaključke in funkcije.

Obliko aplikacije lahko ustvarjamo preko grafičnega vmesnika in s klikanjem dodamo gradnike ter jih postavimo kamor želimo. Lahko jih pišemo v XAML kodi, ali pa v C# kodi. Vse je odvisno od potrebe in želja uporabnika.

C# je objektno usmerjen programski jezik, ki ga je razvilo podjetje Microsoft na osnovi C++. Prevajalnik prevede programsko kodo v jezik neodvisen od platforme, nato pa se programska koda prevede v strojno kodo. Za to skrbi sistem CLR, ki je tudi odgovoren za izvajanje .NET programov ter upravljanje s pomnilnikom.

```

phone:PhoneApplicationPage
  x:Class="Parkomat.MainPage"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
  xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="720"
  FontFamily="{StaticResource PhoneFontFamilyNormal}"
  FontSize="{StaticResource PhoneFontSizeNormal}"
  Foreground="{StaticResource PhoneForegroundBrush}"
  SupportedOrientations="Portrait" Orientation="Portrait"
  xmlns:controls="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls"
  shell:SystemTray.IsVisible="True"
  xmlns:toolkit="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls.Toolkit"
  xmlns:my="clr-namespace:Microsoft.Phone.Controls.Maps;assembly=Microsoft.Phone.Controls.Maps">

<phone:PhoneApplicationPage.Resources>
  <ControlTemplate TargetType="my:Pushpin" x:Key="PinTemplate">
    <Grid>
      <Ellipse Fill="Red" Width="30" Height="30" />
      <Ellipse Fill="Yellow" Width="20" Height="20" />
    </Grid>
  </ControlTemplate>
</phone:PhoneApplicationPage.Resources>

<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="Transparent">
  <!--Pivot Control-->
  <controls:Pivot Title="Parkomat">
    <!--Pivot item one-->
    <controls:PivotItem Header="Maps">
      <Grid>
        <my:Map Height="597" Name="Map" VerticalAlignment="Top"
        CredentialsProvider="App://schemas.microsoft.com/winfx/2006/xaml/presentation;assembly=Microsoft.Phone.Controls.Maps;resource=MapCredentialsProvider"
        CopyrightVisibility="Collapsed"
        LogoVisibility="Collapsed"
        ZoomBarVisibility="Visible">
        </my:Map>
      </Grid>
    </controls:PivotItem>
    <!--Pivot item two-->
    <controls:PivotItem Header="Tools">
      <Grid>
        <ListBox Name="StationList" Margin="0,144,0,0">
          <ListBox.ItemTemplate>
            <DataTemplate>
              <Button Width="Auto" Height="120" Padding="0,0,0,0">

```

Slika 3.4: XAML

3.4 XAML

XAML je označevalni jezik uporabljen za izdelavo uporabniških vmesnikov za aplikacije, ki temeljijo na .NET. Z njim definiramo povezavo med gradniki in podatki, elemente uporabniškega vmesnika, dogodke in ostale stvari. Uporablja se tudi za zapis animacij in vektorske grafike.

Ustvaril ga je Microsoft. Vse kar je ustvarjeno v XAML-u je lahko izraženo z bolj tradicionalnim .NET jezikom, kot je C# ali Visual Basic.NET. Vendar je cilj te tehnologije, da se zmanjša kompleksnost orodij za procesiranje XAML-a

Poglavje 4

Aplikacija

Predstavljena aplikacija omogoča uporabniku pregled parkirnih hiš po Sloveniji in informacije o njih, kot so prosta parkirna mesta, delovni čas in urna postavka. Lahko vklopi navigacijo, ki uporabnika vodi do izbrane parkirne hiše in si shrani njegovo lokacijo, da jo ob vrnitvi najde.

4.1 Dostop do podatkov

4.1.1 Lokalni podatki

Za hitro in gladko delovanje aplikacije je najboljša, da so podatki na voljo takoj, kar pa je možno le takrat, ko so podatki shranjeni lokalno na disku, kjer se aplikacija nahaja. Aplikacija ob prvi uporabi naloži podatke na disk, zato da jih ob naslednjih dostopih hitreje pridobi in uporabi. Podatke pridobi iz spletne strani.

4.1.2 Lokacije parkirnih hiš

Lokacije se za prikaz delovanja aplikacije naložijo iz kode na disk ter se berejo od tam. S tem smo hoteli prikazati, kako bi se ti podatki shranili, če bi jih pridobili iz spleta. Ko so podatki shranjeni na disku, ob vsakem zagonu preveri, če obstaja datoteka s temi podatki. Če obstaja, jo preskoči in doda

podatke v gradnik, preko katerega je vidno, katere parkirne hiše so na voljo. Shranjeni so tudi podatki o ceni, koliko je parkirnih mest in delovni čas.

4.1.3 Podatki o trolah

Ob prvem zagonu se aplikacija poveže na spletno stran, iz katere pridobi vsa imena postaj v Ljubljani ter jih shrani na disk. Ob vnašanju postaje samodejno prikaže celotno ime postaje. To uporabniku olajša vnos in hkrati zmanjša možnost napačnega vnosa, saj s klikom na spustni seznam lahko izbere postajo, ki jo je začel vnašati.

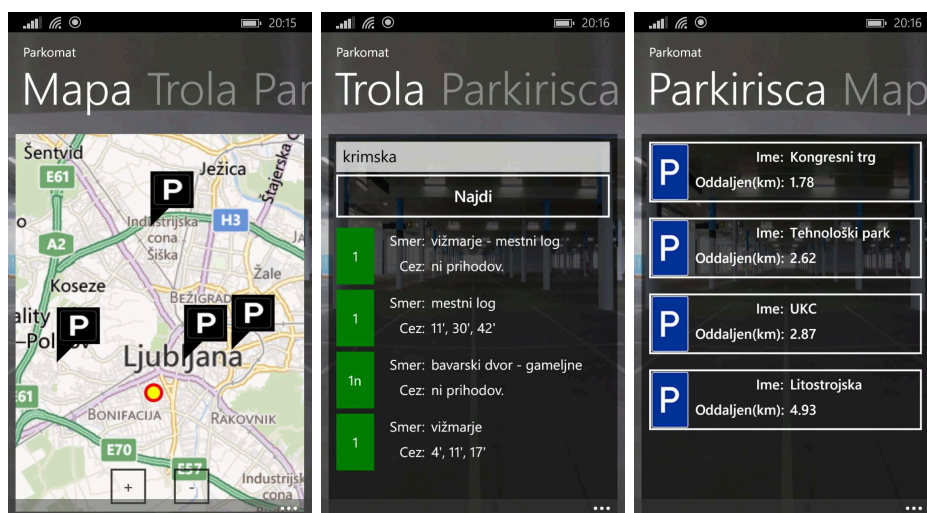
Primer kode, s katero aplikacija pridobi in shrani na disk vse postaje v Ljubljani, je zapisan v nadaljevanju. Podatki so iz spletne strani Mestnega potniškega prometa. Ta koda se zažene že ob prvem zagonu aplikacije in nato ob vsakem iskanju ponudi imena postaj, ki so na voljo. Če ni povezave s spletom se aplikacija ustavi ter opozori uporabnika, naj jo omogoči in še enkrat zažene aplikacijo.

```
{
    progressBar1.IsIndeterminate = true;
    IsolatedStorageFile myIsolatedStorage =
        IsolatedStorageFile.GetUserStoreForApplication();
    client.DownloadStringAsync
        (new Uri("http://bus.talktrack.com/default.aspx/"));
    client.DownloadStringCompleted += new
        DownloadStringCompletedEventHandler(nalozisStran);
    this.autoCompleteBox1.KeyUp += new KeyEventHandler(enterTipka);
}
else
{
    MessageBox.Show("No connection. Connect to
        internet and then reopen application");
}
```


Ko so podatki o postajah pridobljeni iz spletne strani, se morajo še shraniti na disk in dodati v gradnik, ki omogoča, da ponudi besede katere uporabnik delno vpisuje. Ustvarjena je datoteka `postaje.txt`, kjer so shranjena imena postaj. Ob prvi uporabi koda preveri, če ta datoteka morebiti že obstaja, v nasprotnem primeru pa se ustvari in napolni z imeni, hkrati se pa doda v gradnik samodejno zaključevanje. Ko datoteka že obstaja, koda preskoči vnos v datoteko, jo prebere ter vnese imena v gradnik.

```
IsolatedStorageFile ISF =
IsolatedStorageFile.GetUserStoreForApplication();
if (!ISF.FileExists("postaje.txt"))
{
    MessageBox.Show("Samo ob prvi uporabi traja dlje,
    da se naložijo imena postaj v telefon.");
    using (StreamWriter SW = new StreamWriter(new
    IsolatedStorageFileStream("postaje.txt", FileMode.Create,
    FileAccess.Write, ISF)))
    {
        while (a != -1)
        {
            a = vsebina.IndexOf("<option value=", a + 1);
            postaje(a, vsebina, zamik);
        }
        if (listBox2.Items.Count > 0)
            listBox2.Items.RemoveAt(listBox2.Items.Count - 1);
        for (int i = 0; i < listBox2.Items.Count; i++)
        {
            SW.WriteLine(listBox2.Items.ElementAt(i));
        }
        SW.Close();
    }
}
```

```
        this.autoCompleteBox1.ItemsSource =  
            listBox2.Items.ToList();  
    }  
}  
else  
{  
    listBox2.Items.Clear();  
    using (StreamReader reader = new StreamReader(new  
        IsolatedStorageFileStream("postaje.txt", FileMode.Open,  
        FileAccess.Read, ISF)))  
    {  
        while (!reader.EndOfStream)  
            this.listBox2.Items.Add(reader.ReadLine());  
    }  
    this.autoCompleteBox1.ItemsSource = listBox2.Items.ToList();  
}
```

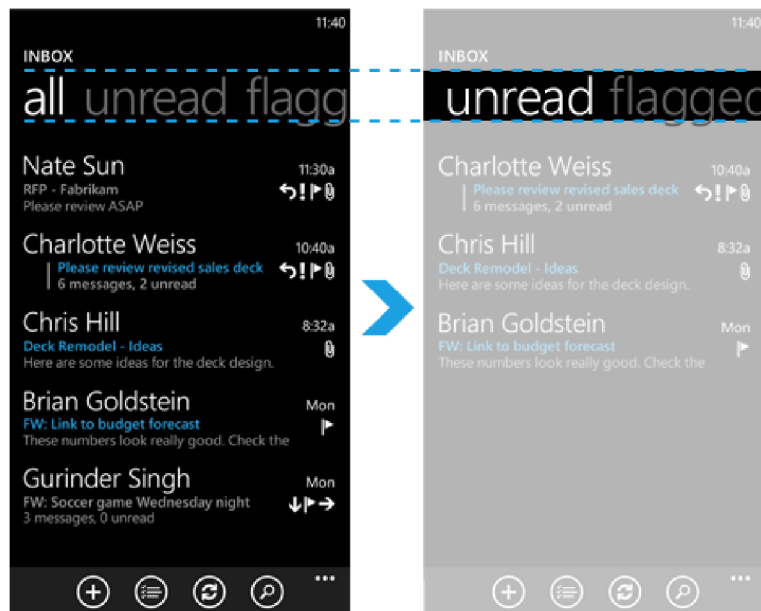


Slika 4.1: Izgled aplikacije

4.2 Aplikacija

Na začetku aplikacije se pokaže zemljevid, kjer so naložene lokacije parkirnih hiš. Ta gradnik je prikazan v Pivot Item. V aplikaciji je pet Pivot Itemov, tri so na začetni strani in lahko premikamo med njimi z drsenjem v levo ali desno. Druga dva sta znotraj aplikacije, do katerih se dostopa s klikom na gumb, prikažeta pa tudi dodatne podrobnosti. Tri glavni meniji so: Mapa Trola Parkirišča

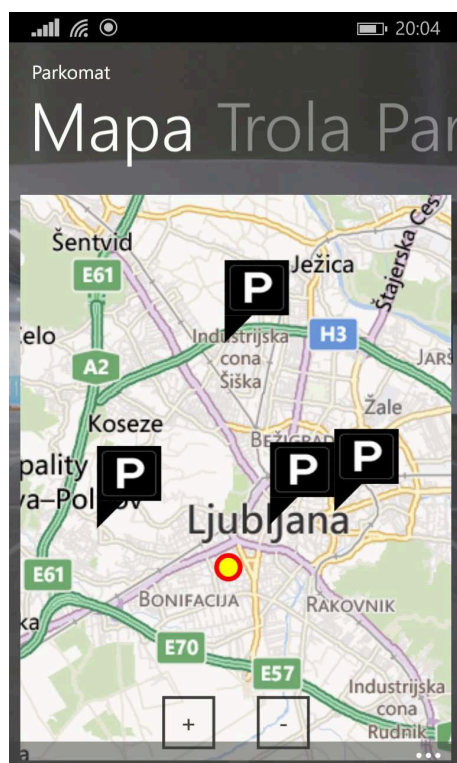
V aplikaciji je tudi aplikacijska vrstica z gumbi, ki še dodatno pokažejo stvari, ki niso takoj potrebne. S klikom na spodnjo vrstico se ta odpre in prikaže dodatne gumbе.



Slika 4.2: Pivot Item

4.2.1 Pivot

V naslovnici gradnika je navigacijska vrstica, ki je obvezna za premikanje med stranmi. Premikanje se izvaja s klikom na naslednje ime ali z drsenjem čez zaslon. Znotraj vsake strani je druga vsebina. Omogoča tudi nastavitev, kdaj naj se vsebina naloži. To se lahko zgodi takoj ob naložitvi aplikacije ali pa ob prikazu strani. S tem se aplikacija odpre hitreje, če niso vsi podatki potrebni ob začetku. Vgrajeno ima tudi premikanje z drsenjem, zato ni potrebno uvajanja.



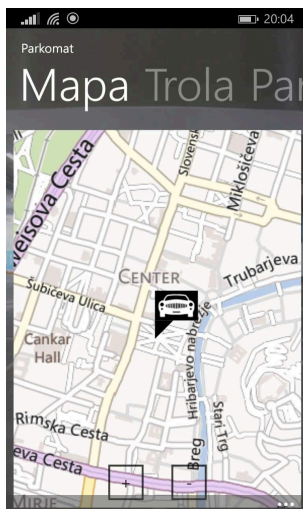
Slika 4.3: Mapa

4.3 Mapa

Na strani Mapa je gradnik zemljevid, ki je osrednji del aplikacije, na katerega se navezujejo drugi deli aplikacije. Za uporabo gradnika zemljevid je potrebna prijava na Microsoftovi spletni strani, s čimer pridobimo ključ za uporabo zemljevida. V nasprotnem primeru se izpisuje besedilo na sredini zaslona in onemogoča pravilno uporabo. Ključ se vnese pri dodajanju gradnika v aplikacijo v XAML.

4.3.1 Prikaz lokacij na zemljevidu

Lokacije na zemljevidu so prikazane s podobo 'žebličkov' (ang. pushpin), ki prikazujejo mesto parkirnih hiš. S klikom na 'žebliček' se odpre pojavno okno, ki ponudi možnosti na voljo za izbiro. Prikažejo se štiri gumbi. Prvi je



Slika 4.4: Gradnik žebliček (ang. pushpin)

za navigacijo do izbrane parkirne hiše. Drugi je za začetek štopanja. Temu sledi gumb 'plačaj', ki odpre novo stran, kje se prikaže izračun za čas, ko je bilo vozilo parkirano v parkirni hiši ter sestavi številko za poklicati, za plačilo parkirnega listka na avtomatu. Ob končanem klicu se števec za čas ponastavi in prikažejo se ostale parkirne hiše. Spodaj je še gumb 'zapri', ki s pritiskom nanj zapre pojavno okno in naprej kaže zemljevid. Navigacija do parkirne hiše se odpre v drugi aplikaciji, ki skrbi za pravilno navigacijo od lokacije na kateri se nahajamo do ciljne točke. Prikaže zemljevid s potjo in seznam z ulicami, kjer je potrebno zaviti. Aplikacijo se lahko zapre in lokacija vozila ostane shranjena na zemljevidu, zato ni potrebno imeti odprte aplikacije in tako višati porabe baterije.

4.3.2 Gradnik žebliček (Pushpin)

Ta gradnik se uporablja za prikaz lokacije na zemljevidu. Glavni argumenti so oznaka, ime, slika in lokacija. Na zemljevidu se prikaže kot oznaka in puščica spodaj prikazuje točno lokacijo. Lokacija vsebuje dva podatka, in sicer širino ter dolžino. Na spodnji sliki je prikazan izgled na zemljevidu ter

koda, potrebna za prikaz.

```
Pushpin dat = new Pushpin();
dat.Name = "Park";
dat.Tag = name + ":" + cost + ":" + price;
dat.Location = new GeoCoordinate(lati, longi);
ImageBrush image = new ImageBrush()
{
    ImageSource = new System.Windows.Media.Imaging.BitmapImage
        (new Uri("/Images/Park.png", UriKind.Relative))
};

dat.Content = new Rectangle()
{
    Fill = image,
    StrokeThickness = 1,
    Height = 50,
    Width = 60
};

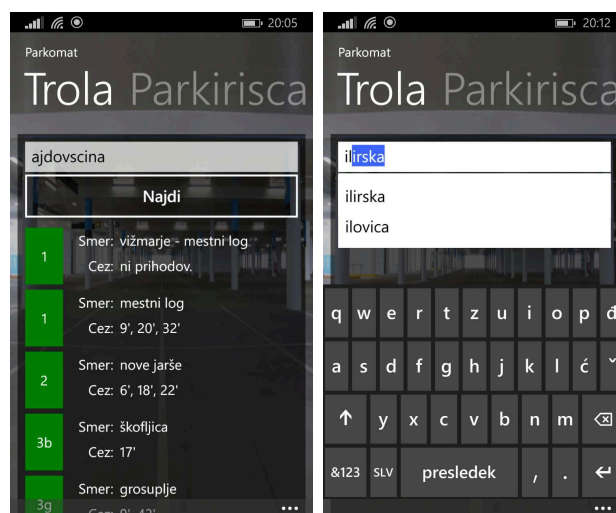
Map.Children.Add(dat);
dat.MouseLeftButtonUp +=
new MouseButtonEventHandler(pushPin_Tap);
```

Ustvari se nov element Pushpin, z imenom Park in oznako ime parkirišča, čas in cena. Nato se vnese lokacijo, ki je prebrana iz datoteke, shranjene na disku. Doda se mu slika, ki je že vnaprej določena in nato se doda na zemljevid. Na koncu se doda tudi funkcija, ki se sproži na klik.

4.3.3 Dodatne funkcije zemljevida

Ob uporabi zemljevida, kjer se prikažejo lokacije parkirnih hiš, je na voljo še nekaj funkcionalnosti, ki omogočajo uporabniku, da spremeni izgled zemlje-

vida ali posreduje podatke o svoji lokaciji. S pritiskom simbola na skrajnem spodnjem desnem robu ekrana (tri pikice) se prikažejo novi gumbi, ki ob pritisku prikažejo trenutno lokacijo ter spremenijo izgled zemljevida iz satelitskega v grafičnega in nazaj. Lahko tudi pošlje svojo lokacijo preko kratkega sporočila ali elektronske pošte ter nastavi svojo lokacijo na zemljevidu, ki se nato shrani. S klikom na gumb 'najdi me' se zemljevid postavi na sredino lokacije, kjer se uporabnik nahaja.



Slika 4.5: Trola

4.4 Trola

Na strani, kjer piše 'Trola', si lahko uporabnik ogleda prihod avtobusov glede na iskano postajo. Ob prvem zagonu aplikacije se imena postaj naložijo iz spletne strani v disk na telefonu in ostanejo shranjena za kasnejšo uporabo. To omogoča, da lahko aplikacija sama dokonča vnos imena, ki ga uporabnik tipka. Imena postaj se shranijo v ListBox, ki je kot seznam, le da lahko na več načinov manipuliramo z njim.

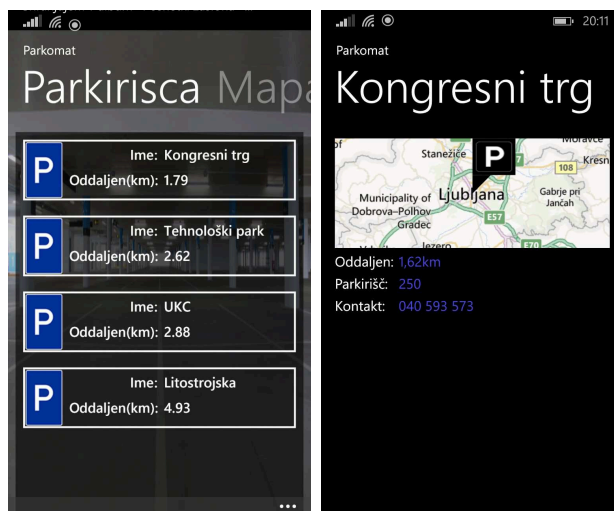
Ko uporabnik vnese želeno postajo, pritisne tipko 'enter' na tipkovnici ali pritisne gumb 'najdi'. Glede na vnesen tekst se aplikacija poveže na spletno stran www.trola.si, kjer lahko vidi, katere številke avtobusov so vezane na to postajo in tri naslednje čase v minutah.

Prikaz časov in številke avtobusov je prikazan v ListBoxu, kjer se dodajo te podatki v gumb. Znotraj gumba so te podatki vezani na določen del, tako vedno izpiše pravilne podatke na pravilnem mestu.

```

<ListBox.ItemTemplate>
    <DataTemplate>
        <StackPanel Orientation = "Horizontal" Height = "90"
            Margin = "12,4,12,4">
            <Grid Height = "90" Width = "60" Background = "Green" >
                <TextBlock Margin = "10, 0, 10, 0" FontSize = "22"
                    Text = "{Binding IdStation}" Height = "30"
                    HorizontalAlignment = "Center" />
            </Grid>
            <StackPanel Orientation = "Vertical" Height = "80"
                Margin = "10, 0, 0, 0">
                <StackPanel Orientation = "Horizontal" Height = "40">
                    <TextBlock Width = "70" TextAlignment = "Right"
                        Margin = "0, 0, 10, 0" FontSize = "22" Text = "Smer:" Height =
                    <TextBlock Width = "400" FontSize = "22"
                        Text = "{Binding StationName}" Height = "40" />
                </StackPanel>
                <StackPanel Orientation = "Horizontal" Height = "40">
                    <TextBlock Width = "70" TextAlignment = "Right"
                        Margin = "0, 0, 10, 0" FontSize = "22" Text = "Cez:" Height =
                    <TextBlock Width = "400" FontSize = "22"
                        Text = "{Binding Time}" Height = "40" />
                </StackPanel>
            </StackPanel>
        </StackPanel>
    </DataTemplate>
</ListBox.ItemTemplate>
</ListBox>
<toolkit:AutoCompleteBox HorizontalAlignment = "Center"
    Name = "autoCompleteBox1" VerticalAlignment = "Top"
    Width = "456" DataContext = "{Binding}" IsTextCompletionEnabled = "True" />

```



Slika 4.6: Parkirišča

4.5 Parkirišča

Na strani Parkirišča si lahko uporabnik ogleda parkirna mesta razvrščena po oddaljenosti od njegove trenutne lokacije. V vsakem gumbu se prikaže ime parkirne hiše in oddaljenost v kilometrih. S klikom na gumb se odpre novo okno, kjer si lahko pogleda želeno parkirno hišo na zemljevidu ter njene podrobne podatke kot so ime, število parkirnih mest, delovni čas in kontaktno številko.

Tukaj sta tudi uporabljena gradnika `ListBox` in gumb za prikaz podatkov kot pri prikazu avtobusov. Ti podatki so pridobljeni iz lokalnega diska, kjer so bili že prej podatki shranjeni.

```
<ListBox Name = "listBox3" HorizontalAlignment = "Left" Height = "607"
VerticalAlignment = "Top" Width = "456" >
<ListBox.ItemTemplate>
    <DataTemplate>
        <Button Width = "Auto" Height = "120" Padding = "0, 0, 0, 0"
Click = "showInfo" >
        <Button.Content>
```

```

<StackPanel Orientation = "Horizontal" Height = "90">
    <Grid Height = "90" Width = "60" >
        <Grid.Background>
            <ImageBrush Stretch = "None"
                ImageSource = "/Images/parking.png" />
        </Grid.Background>
    </Grid>
    <StackPanel Orientation = "Vertical" Height = "80"
        Margin = "10, 0, 0, 0">
        <StackPanel Orientation = "Horizontal" Height = "40">
            <TextBlock Width = "140" TextAlignment = "Right"
                Margin = "0, 0, 10, 0" FontSize = "22" Text = "Ime:"
                Height = "40" />
            <TextBlock Width = "320" FontSize = "22"
                Text = "{Binding ParkingName}" Height = "40" />
        </StackPanel>
        <StackPanel Orientation = "Horizontal" Height = "40">
            <TextBlock Width = "140" TextAlignment = "Right"
                Margin = "0, 0, 10, 0" FontSize = "22" Text = "Oddaljen(km):" Height = "40" />
            <TextBlock Width = "320" FontSize = "22"
                Text = "{Binding Dist}" Height = "40" />
        </StackPanel>
    </StackPanel>
</StackPanel>
</Button.Content>
</Button>
</DataTemplate>
</ListBox.ItemTemplate>
</ListBox>

```

In ustvarjen razred za parkirišča

```
namespace Parkomat
```

```
{
    class Parking
    {
        public int IdPark
        { get; set; }

        public string ParkingName
        { get; set; }

        public double Lat
        { get; set; }

        public double Lon
        { get; set; }

        public double Dist
        { get; set; }

        public string AllParkings
        { get; set; }

        public Parking(int idPark, string parkingName, double lat, double lon, double dist)
        {
            this.IdPark = idPark;
            this.ParkingName = parkingName;
            this.Lat = lat;
            this.Lon = lon;
            this.Dist = dist;
            this.AllParkings = allParkings;
        }
    }
}
```

}

4.6 Parkirne hiše

V tem projektu bi morale sodelovati tudi parkirne hiše, vendar je aplikacija nastajala brez njih in je trenutno prototip. Lokacije, ki so prikazane na zemljevidu, so dodane ročno glede na prejšnjo poznavanje njihovih lokacij. Prikazano je, kako deluje in če bi lastniki parkirnih hiš samo vnesli te podatke, bi aplikacija delovala tudi z njihovimi lokacijami.

V ta namen bi ustvaril še spletno stran, kjer bi se lahko registrirali in vnesli podatke za prikaz v aplikaciji. Težko bi bilo sicer pridobiti podatke o trenutni zasedenosti parkirne hiše, saj vse ne beležijo število parkiranih vozil oziroma ne pošiljajo teh podatkov na splet, kjer bi lahko dostopal do njih.

4.7 Vključene knjižnice

4.7.1 Windows Phone Toolkit in Windows Phone Map

Windows Phone Toolkit je skupek dodatnih gradnikov, ki nam olajšajo sam razvoj in nam vizualno polepšajo aplikacijo. Orodje je brezplačno.

Izmed ponujenih gradnikov, ki jih vsebuje ta dodatek smo uporabili AutoCompleteBox oziroma okno za samodejno zaključevanje besed.

Windows Phone Map je knjižnica, ki omogoča prikaz in delo z zemljevidom in lokacijo.

Uporabili smo funkcije za prikaz trenutne lokacije uporabnika in dodajanje lokacij na zemljevid. Uporabili smo tudi funkcijo za izračun razdalje med uporabnikom in parkirno hišo.

4.8 Testiranje

Aplikacijo smo redno testirali, zato da smo sproti odpravljali napake, ki so se pokazale in tudi videli delovanje in prikaz podatkov. Testiranje smo opravljali na telefonu, ki ga je dal v uporabo mentor, prof. Rok Rupnik, in na emulatorju, ki omogoča testiranje na več kot enem telefonu.

Emulator omogoča virtualno okolje, v katerem lahko testiramo delovanje WP aplikacij. Znotraj njega je več različnih verzij sistemov, ki nam omogočajo tudi testiranje na starejših različicah in različnih ločljivostih zaslona. Emulator omogoča večtočkovni dotik, obračanje in tresenje. Poleg naštetega pa omogoča tudi delanje posnetkov zaslona, za objavo aplikacije v trgovini in pregled lokacije.

4.9 Deljenje aplikacije

Nameščanje aplikacij na telefon je uradno možno samo preko trgovine Windows Store, s čimer omejujejo zlorabo uporabnikovih podatkov. Vse aplikacije so, preden jih objavijo in preverijo, testirane če morebiti vsebujejo škodljivo kodo, ki bi oškodovala uporabnika. Za objavo aplikacije v trgovini je potrebno imeti odprt račun, le ta, razen za študente, plačljiv. Pri oddajanju aplikacije moramo napisati podrobnosti o sami aplikaciji ter dodati slike, ki so prikazane. Po želji pa se lahko tudi omeji, v katerih državah naj bo na voljo.

Ko je aplikacija uspešno dodana v trgovino in objavljena, je možno preverjati podrobnosti o njej. Prikaže se, kolikokrat je bila naložena na telefon, koliko uporabnikov jo ima trenutno naloženih ter kolikokrat se je aplikacija ustavila med samim delovanjem. Vse to omogoča, da se aplikacijo optimizira in izboljša. Prikazani so tudi komentarji, ki so jih napisali uporabniki po testiranju. Ob komentiranju lahko tudi ocenijo aplikacijo in s tem podajo svoje mnenje. Če so kakšne napake v aplikaciji, se jih hitro popravi in spet naloži na trgovino, ta postopek je hitrejši ter traja do dva dneva.

Poglavje 5

Sklepne ugotovitve

Med samim ustvarjanjem aplikacije so se nam sproti pojavljale nove ideje, kako še izboljšati in povečati aplikacijo. Ker je bilo vseh le preveč, smo si te ideje raje zapisal in jim bomo lahko vključili v naslednji posodobitvi. Najprej bi bilo najboljše slišati/videti odzive in mnenja uporabnikov.

Aplikacijo bi hotel razširiti iz prototipa v uporabno aplikacijo. Da bi to bilo realizirano, bi morali sodelovati še lastniki parkirišč, ki bi dodali svoje podatke. Tako bi se baza parkirišč samodejno širila in omogočala uporabnikom dostop do novih parkirnih mest preko aplikacije.

Plačilo bi potekalo preko aplikacije, uporabnik bi plačal s klicem na številko in dobil nazaj bar kodo, s čimer bi lahko zapustil parkirišče. S tem ni potrebno po parkomatih. Da bi se uporabniki, ki so parkirali vozilo na parkirišču zapisali v bazo in bi lahko s tem videl koliko parkirnih mest je še prostih.

Da bi lahko rezerviral parkirno mesto od doma in bi te čakalo ko bi prišel v parkirno hišo. Ko bi označil rezervacijo, bi se ti začelo računat parkiranje.

Izpis časa parkiranja v Live Tile ploščici na namizju, da ne bi bilo potrebno zaganjat aplikacijo.

Slike

3.1	Adobe Photoshop CS6 verzija.	6
3.2	Windows Phone 8.1	7
3.3	Visual Studio Express for Windows Phone	10
3.4	XAML	11
4.1	Izgled aplikacije	17
4.2	Pivot Item	18
4.3	Mapa	19
4.4	Gradnik žebliček (ang. pushpin)	20
4.5	Trola	23
4.6	Parkirišča	25

Literatura

- [1] MSDN: Map control for Windows Phone. Dostopno na:
[http://msdn.microsoft.com/en-us/library/windows/apps/jj207045\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/jj207045(v=vs.105).aspx)
- [2] Nokia: Guide to the Windows Phone 8 Maps API. Dostopno na:
<http://developer.nokia.com/resources/library/Lumia/change-history/archived-content/maps-and-navigation/guide-to-the-wp8-maps-api.html>
- [3] MSDN: Create settings page. Dostopno na:
[http://msdn.microsoft.com/en-us/library/windows/apps/ff769510\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/ff769510(v=vs.105).aspx)
- [4] MSDN: Pivot control for Windows Phone. Dostopno na:
[http://msdn.microsoft.com/en-us/library/windows/apps/ff941098\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/ff941098(v=vs.105).aspx)
- [5] MSDN: StreamWriter Class. Dostopno na:
[http://msdn.microsoft.com/en-us/library/windows/apps/system.io.streamwriter\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/system.io.streamwriter(v=vs.105).aspx)
- [6] MSDN: WebClient Class. Dostopno na:
[http://msdn.microsoft.com/en-us/library/windows/apps/system.net.webclient\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/system.net.webclient(v=vs.105).aspx)
- [7] MSDN: Windows Phone Emulator. Dostopno na:
[http://msdn.microsoft.com/en-us/library/windows/apps/ff402563\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windows/apps/ff402563(v=vs.105).aspx)
- [8] Adding text to Pushpin via C#. Dostopno na:
<http://stackoverflow.com/questions/2805355/silverlight-adding-text-to-pushpin-in-bing-maps-via-c-sharp>

- [9] Wikipedija o Windows Phone. Dostopno na:
http://en.wikipedia.org/wiki/Windows_Phone

- [10] Wikipedija o Visual Studio. Dostopno na:
http://en.wikipedia.org/wiki/Microsoft_Visual_Studio